https://doi.org/10.2352/ISSN.2169-2629.2017.32.32
Reprinted from
Journal of Imaging Science and Technology® 60(5): 050407-1–050407-9, 2016.

Orientation Modulation for Data Hiding in Chrominance Channels of Direct Binary Search Halftone Prints

Vlado Kitanovski▲ and Marius Pedersen▲
The Norwegian Colour and Visual Computing Laboratory, Faculty of Computer Science and Media Technology, NTNU - Norwegian University of Science and Technology, Gjøvik, Norway
E-mail: vlado.kitanovski@ntnu.no

Abstract. *In this article, we propose a joint halftoning and data hiding technique for color images. To ensure high quality of the printed image, the color direct binary search (CDBS) iterative halftoning algorithm is used. The proposed approach uses the commonly available cyan, magenta and yellow colorants to hide data in the chrominance channels. Orientation modulation is used for data embedding during the iterative CDBS halftoning stage. The detector is using PCA-learned components to extract the embedded data from the scanned image. Experimental results show that this proposed CDBS-based data hiding method offers both higher data hiding capacity and higher robustness to the print-and-scan channel when compared to the state-of-the-art grayscale counterpart method. The relatively high correct detection rate make this approach suitable for applications which require exact extraction of embedded data in prints.*

INTRODUCTION

Despite the expansion of digital media, printed content will be used in the foreseeable future. Large number of official documents such as birth and citizenship certificates, identification cards, passports and educational diplomas still use the printed format. Magazines, posters, billboards, or product packaging, are just other examples of commonly used printed media. For many of these prints there is an increasing need for technologies that allow data to be embedded into the printed content, preferably in an imperceptible way, so that the data can be subsequently extracted using suitable imaging devices – such as scanners and consumer cameras. This approach for data hiding, also known as watermarking,[1] provides additional functionalities for security applications like document authentication, tamper detection, source control, or steganography. Other than security applications, hiding data can be used in general to create "smart" prints[2] where the embedded data may give additional information about the printed content, or point to an URL where the user can explore more options related to the print.

Printers are typically binary devices which either deposit ink or not at a given location on the printing media. Prior to printing, digital images are binarized per colorant using a technique called halftoning, which exploits increased spatial resolution of printed dots to compensate for the lost amplitude resolution due to the binarization[3] (depositing ink or not). Halftoning can cause severe distortion to the embedded data if it is not taken into account during the data embedding stage; this is the main reason for developing a wide range of watermarking techniques which embed the data during the halftoning process. This joint halftoning and watermarking approach implicitly includes some form of adaptation to the print-and-scan channel, in order to achieve higher robustness of the embedded data.

When developing data hiding techniques for printed images, there are several challenges to be tackled. The most important challenge is achieving robustness to the print-and-scan channel—the process of data extraction from prints means that the data inevitably pass through this channel. Apart from the heavy binarization (halftoning), other sources of distortion in the print-and-scan channel include: color space conversion (gamut mapping), physical and optical dot gain, non-uniform lattice of printed dots, and global geometric distortions (scaling, rotation, perspective projection, or barrel distortion). Addressing these sources of distortion either by prior modelling or posterior compensation would in general increase the overall robustness of hidden data.[4] Another important challenge is the printed image quality and watermark imperceptibility. The embedded data should normally be visually imperceptible, and the visual quality of the watermarked printed image should be very close, ideally identical, to the quality of the non-watermarked printed image. The amount of embeddable data (capacity) with certain level of reliable recovery is also an important feature to be considered when designing data hiding systems. The amount of embeddable (and recoverable) data per unit area should meet the application-specific requirements. High correct detection rate (CDR) is always desirable, and for some applications it is critical, e.g. where every single bit of the embedded data carries equally significant information and needs to be correctly recovered. Error correction coding (ECC) techniques can guarantee error-free data extraction only above a certain level of raw CDR. The security of the hidden data is also an important issue to be addressed—the designed system needs to be secure against application-specific attacks.

▲ IS&T Members.

Received Apr. 9, 2016; accepted for publication June 2, 2016; published online July 12, 2016. Associate Editor: Yeong-Ho Ha.

In its early stages the research work on combining halftoning and data embedding was focused on monochromatic printing of grayscale images. Later the research work was extended to color images; however, the amount of published work utilizing color images is significantly lower. In the early work, Fu and Au proposed a technique for data embedding in black and white halftones by smart pixel toggling, which is compensated by an opposite toggle in the pixel neighborhood to preserve the average gray level.[5] A further improved technique was proposed by Guo where a filter modelling the human visual system (HVS) is used to obtain the optimal neighboring pixel whose change will result in minimal visual error.[6] The smart pixel toggling idea was extended to use error diffusion halftoning algorithm, where a non-causal kernel is used to reduce the error from the pixel toggling.[7] Another data hiding technique based on pixel toggling while minimizing halftoning error metric was proposed by Pei *et al*[8] Their method for embedding a visual pattern in two or more error-diffused halftones is suitable for color halftones. The data embedding is performed by forcing conjugacy between pixels of different halftones. The detection is visual—when overlaying the halftones the hidden visual pattern appears due to the enforced conjugacy. However, all of these approaches are embedding a data bit in a single halftone dot, so their print-and-scan robustness is relatively low. Pei and Guo proposed a kernel-alternated data hiding technique in error diffusion halftones.[9] The idea is to use two different kernels for error diffusion according to the binary to-be embedded watermark bits. The host image is block segmented so that each block carries one watermark bit and is halftoned using the corresponding kernel. The detection can be performed in spatial domain by using look-up tables, or in frequency domain as the two kernels used, Jarvis[10] and Stucki,[11] have different frequency responses. Oztan and Sharma proposed a data hiding technique[12] for clustered-dot halftones based on continuous phase modulation (CPM). The phase of the clustered dots is altered according to the bit from the visual watermark pattern. The detection is visual, by overlaying the CPM halftone with non-modulated one. The same authors extended this idea to bidirectional CPM.[13] The phase shift of halftone dots is performed in two different directions (e.g. horizontal and vertical), so that two different visual watermarks can be embedded in the same place without interfering with each other. Another proposed extension of CPM by Oztan and Sharma was to color halftone screens.[14] The colorant CMYK screens are shifted from each other by appropriate angles to minimize dot overlap and moiré effects. A technique for estimation of the colorant screens from RGB scans is also proposed. The detection is visual—by overlaying appropriate non-phase-modulated halftone screen onto each of the colorant screens. Bulan et al. proposed a data hiding technique for clustered-dot halftones called orientation modulation.[15] It is using the orientation of the elliptically shaped clustered dots to embed data. The detection is computational, and is using image moments to detect the dot orientation. This approach achieves high robustness to the print-and-scan channel, and it also offers high data hiding capacity. This technique was extended to color barcodes,[16] but not to images with arbitrary content.

Few data hiding techniques have been proposed which are based on the direct binary search (DBS) halftoning algorithm.[17,18] DBS is an iterative halftoning technique which gives the best image quality due to the utilization of HVS models in the halftone texture optimization stage. Wan proposed a simple halftone pixel toggling approach followed by DBS to improve the halftone quality.[19] Zhuge proposed a data hiding technique which is using two DBS halftoned images to hide visual watermark.[20] This approach is similar to the "conjugacy approach"—the data embedding is performed by enforcing the levels of co-located halftone pixels to correspond to the pixels of the watermark. The minimization is performed jointly on both halftone images and the detection is visual by overlaying the two DBS halftones. Kacker and Allebach incorporated a block-based spread-spectrum data embedding in the iterative DBS technique.[21] An HVS-based metric is used to estimate the error between the continuous and the halftone image, and another measure is used to estimate the quality of watermark detection. While this technique is robust to the print-and-scan channel, it is a 0-bit watermarking scheme, meaning that the watermark carries no useful information – the detector can only decide if a watermark signature is present or not. Guo et al. used the orientation modulation approach to hide data in DBS halftoned images.[22,23] The data is embedded by modulating the orientation of the point-spread function which is used as an HVS-based filter in the DBS halftoning process. The detector is using trained filters and naive Bayesian classifier to detect the orientation features in frequency domain. The robustness to the print-and-scan channel is fair and the CDR allows reliable data embedding and detection. An improvement of this technique was proposed,[24] where the embeddable capacity is increased by using multi-layer watermarks, and the computational complexity is decreased by using look-up tables during the DBS halftoning and watermarking process.

In this article, we propose a joint halftoning and data hiding technique based on the CDBS halftoning algorithm.[25] The proposed technique for data embedding is an extension of the previously described grayscale approach[23] to color halftones. There are several possible ways to extend the original DBS-based orientation modulation to color; in this work we are using only the chrominance channels for data embedding as this results in improved capacity and print-and-scan robustness while keeping the visual degradation almost imperceptible. Data is embedded by modulating the orientation of the chrominance point-spread function used in the HVS-model-based iterative CDBS halftoning. The proposed detection technique is computational, and is utilizing learned templates using principal component analysis (PCA) for correlation-based detection of the embedded data from the scans. The obtained evaluation results show that this method is suitable for a wide range of print-and-scan robust data hiding applications.

The next section describes the approach for data embedding during CDBS halftoning. The process of data extractions from scans is presented in Data Extraction from Scanned CDBS Prints, followed by evaluation results in Evaluation and concluding remarks in Conclusion.

DATA EMBEDDING IN CDBS HALFTONES

This section presents the joint halftoning and data embedding approach. The watermark is always assumed to be binary data; in cases where n-ary symbols are embedded, it is assumed that the binary watermark had been converted to n-ary data. Both the continuous and the halftone color image are of the same size, $R \times C$. The data is embedded into non-overlapping blocks of size $M \times N$. The embedding is performed by modifying the orientation (per $M \times N$ block) of the chrominance point-spread function which is used in the CDBS halftoning. The number of different orientations is of the form 2^B, where B is the number of bits embedded in a single block of one chrominance channel. The total number of embedded bits per color halftone block is $2B$. The proposed approach is restricted to bi-level three-colorant (CMY) printers as well as bi-level four-colorant printers with full undercolor removal. The assumed printing geometry is dot-on-dot, the continuous-tone RGB image is normalized to the [0, 1] range, and the possible CMY halftone pixel values are: "1" - ink dot will be printed, and "0" – no ink will be deposited. In this work, we use naive RGB to CMY conversion: $C = 1 – R$, $M = 1 – G$, $Y = 1 – B$. As this work is mainly focused on the data embedding and its visual impact on the halftones, the absence of a full colorimetric printing workflow is assumed to be not critical. Detailed description of the halftoning and the data embedding process is given in the next two subsections.

CDBS Halftoning

The color direct binary search halftoning algorithm[25] is an iterative halftoning algorithm which minimizes the HVS-model-based perceived error between the continuous image and the halftone image. Starting from an initial halftone, every halftone pixel is modified (toggled) and swapped with each of the eight neighbors. The effects of these trial changes (seven possible toggles + eight possible swaps for a 3-channel halftone) are evaluated so that only the change (if any) which minimizes the perceived error the most is accepted. All halftone pixels are examined in several iterations until any possible pixel change does not reduce the perceived error. Depending on the image this convergence normally happens after 15–40 iterations; however, using our implementation very high quality and almost converged CDBS halftone is typically obtained only after 10 iterations, regardless of the image content.

The original CDBS is using $Y_yC_xC_z$ Ref. 26 as minimization color space, but the mathematical formulation of CDBS in this section is not bound to any color space. We denote the color channels of original continuous image and the halftone image with $f_i[\mathbf{m}]$ and $g_i[\mathbf{m}]$, respectively; $[\mathbf{m}] = [m, n]$ and $(\mathbf{x}) = (x, y)$ are discrete and continuous spatial coordinates. The index i denotes the color channel—it can be any channel from the following sets: {R,G,B}, {C,M,Y}, $\{Y_y, C_x, C_z\}$, $\{Y, C_b, C_r\}$, or another color space. The error image between the continuous and the halftone image is:

$$e_i[\mathbf{m}] = f_i[\mathbf{m}] - g_i[\mathbf{m}]. \tag{1}$$

The perceived error $\tilde{e}_i(\mathbf{x})$ between these two images, using a linear and channel-independent HVS model is given in continuous spatial coordinates:

$$\tilde{e}_i(\mathbf{x}) = \sum_{\mathbf{m}} p_i(\mathbf{x} - \boldsymbol{Xm}) e_i[\mathbf{m}], \tag{2}$$

where $p_i(\mathbf{x})$ is a point-spread function for the ith color channel, the product matrix $\boldsymbol{Xm}$ addresses all of the discrete samples of the error image $e_i[\mathbf{m}]$. The error metric E to be minimized is the sum of squared perceived errors in all color channels:

$$E = \sum_i \int \tilde{e}_i^2(\mathbf{x})\, d\mathbf{x}. \tag{3}$$

It is important to be noted that while the halftone $g_i[\mathbf{m}]$ is normally in CMY space, the error E may be expressed (using Eqs. (1)–(3)) and minimized in another color space, usually a luminance-opponent-color space. In this case, both the continuous and the halftone image need to be converted before calculating the error in the minimization space. It has been shown[18,25] that the error E can be expressed as:

$$E = \sum_i \sum_{\mathbf{m}} e_i[\mathbf{m}] c_{pe,i}[\mathbf{m}], \tag{4}$$

where the matrix $c_{pe,i}[\mathbf{m}]$ is the error $e_i[\mathbf{m}]$ filtered with the flipped matrix $c_{pp,i}[\mathbf{m}]$. This $c_{pp,i}[\mathbf{m}]$ matrix is the autocorrelation function $c_{pp,i}(\mathbf{x})$ of the point-spread function $p_i(\mathbf{x})$ evaluated on the discrete coordinates of the halftone image $\mathbf{x} = \boldsymbol{Xm}$:

$$c_{pe,i}[\mathbf{m}] = \sum_{\mathbf{n}} e_i[\mathbf{n}]\, c_{pp,i}[\mathbf{n} - \mathbf{m}] \tag{5}$$

$$c_{pp,i}[\mathbf{m}] = c_{pp,i}(\mathbf{x} = \boldsymbol{Xm})\,;\; c_{pp,i}(\mathbf{x}) = \int p_i(\mathbf{x}) p_i(\mathbf{y} + \mathbf{x})\, d\mathbf{y}. \tag{6}$$

When a halftone pixel at location denoted as $\mathbf{m}_t$ in the ith (CMY) channel is modified (toggled), the change in the halftone can be denoted with the values $\{a_i[\mathbf{m}_t], 1 \le i \le 3\}$. The value of $a_i[\mathbf{m}_t]$ can be either "-1" –if the ith color channel pixel is switched off, "0" –if the ith color channel pixel is not changed, or "1" –if the ith color channel pixel is switched on. This halftone pixel change will cause pixel value changes in the minimization color space which does not need to be the same as the halftone color space. These changes in the minimization color space, denoted with the values $\{b_i[\mathbf{m}_t], 1 \le i \le 3\}$, can be calculated from $\{a_i[\mathbf{m}_t]\}$ by involving the appropriate color space transform between the halftone color space and the minimization color space. The change in the error E caused by this halftone pixel change is given by:

$$\Delta E_t = \sum_i (b_i^2[\mathbf{m}_t]\boldsymbol{c}_{pp,i}[0] + 2b_i^2[\mathbf{m}_t]\boldsymbol{c}_{pe,i}[\mathbf{m}_t]). \tag{7}$$

When a halftone pixel at location $\mathbf{m}_t$ in the ith (CMY) channel is swapped with a neighboring halftone pixel at location denoted as $\mathbf{m}_s$, the change can be seen as two consecutive toggles defined with the set of values $\{a_i\,[\mathbf{m}_t]\,,\, b_i\,[\mathbf{m}_t]\,,\, a_i\,[\mathbf{m}_s]\,,\, b_i\,[\mathbf{m}_s]\}$. The change in the error E caused by this pixel swap is given by:

$$\begin{aligned}\Delta E_s = \sum_i \Big(& b_i^2[\mathbf{m}_t]\boldsymbol{c}_{pp,i}[0] + 2b_i^2[\mathbf{m}_t]\boldsymbol{c}_{pe,i}[\mathbf{m}_t] \\ &+ b_i^2[\mathbf{m}_s]\boldsymbol{c}_{pp,i}[0] + 2b_i^2[\mathbf{m}_s]\boldsymbol{c}_{pe,i}[\mathbf{m}_s]\Big)\,.\end{aligned} \tag{8}$$

If ΔE is negative, it means the trial change is reducing the error so it is a possible candidate for an accepted change. After all possible toggles and swaps are examined for a given pixel, the change which results in the biggest error reduction (biggest abs (ΔE)) is accepted, and both the halftone and the $\boldsymbol{c}_{pe,i}\,[\mathbf{m}]$ matrix are updated:

$$\begin{aligned} g_{i,new}\,[\mathbf{m}] &= g_i\,[\mathbf{m}] + a_i\,[\mathbf{m}_t]\,\delta[\mathbf{m}-\mathbf{m}_t] + a_i\,[\mathbf{m}_s]\,\delta[\mathbf{m}-\mathbf{m}_s], \\ c_{pe,i,new}\,[\mathbf{m}] &= \boldsymbol{c}_{pe,i}\,[\mathbf{m}] + b_i\,[\mathbf{m}_t]\,\boldsymbol{c}_{pp,i}[\mathbf{m}-\mathbf{m}_t] \\ &\quad + b_i\,[\mathbf{m}_s]\,\boldsymbol{c}_{pp,i}[\mathbf{m}-\mathbf{m}_s], \\ \delta[\mathbf{m}] &= \begin{cases} 1, & \mathbf{m}=0 \\ 0, & \mathbf{m}\neq 0. \end{cases} \end{aligned} \tag{9}$$

Equations (7) and (8) show that no actual point-spread function filtering is needed every time a change is made in the halftone; the impact of the trial pixel change on the error E can be easily calculated from $\boldsymbol{c}_{pp,i}[\mathbf{m}]$ and $\boldsymbol{c}_{pe,i}[\mathbf{m}]$, which are initialized using Eqs. (5), (6) and (1), and $\boldsymbol{c}_{pe,i}[\mathbf{m}]$ is constantly updated using Eq. (9) during the iterations.

The initial CMY halftone for the CDBS algorithm can be either a random halftone, or as originally proposed,[25] a channel-independent DBS halftone which is obtained by halftoning the colorants separately with a monochromatic DBS[18] using a point-spread function which models the luminance contrast sensitivity. The monochromatic DBS is implemented using the same set of Equations (1)–(9), with the only change that both continuous and halftone images are single channel images. Channel-independent DBS is also used as a starting halftone in this work—it results in a faster overall computation when compared to calculating a CDBS halftone from a random initial halftone. The search strategy used in this work is a block-based search strategy:[27] instead of performing trial changes and accepting them on a pixel level, the block-based strategy is performing the trial changes for every pixel, but only one change is accepted within a $D \times D$ block. Accepting only the change which reduces the error the most within a block ensures faster minimization convergence (up to 10 times faster).[27]

Data Embedding using Orientation Modulation

Using orientation modulation for data embedding during iterative DBS halftoning of grayscale images was originally proposed by Guo *et al*[23] The main idea of this technique is to use a point-spread function (in Eq. (6)) which is modified (oriented) according to the data which is to be embedded in the $M \times N$ block. Each different orientation can encode one n-ary symbol. Such a modified point-spread function should no longer be seen as one which models the HVS behavior, but rather as a function which controls the orientation of error distribution during the iterative minimization. The resulting halftone texture is coarser (the error is more spread) in the orientation of the point-spread function (and it is smoother in the perpendicular orientation), which can be easily detected in the frequency domain because of lower energy concentration in the smoothening spatial orientation.

There are different possible ways to extend the original, grayscale-based, point-spread function orientation modulation embedding to CDBS halftones. Our initial empirical test showed that using only chrominance channels for data embedding while leaving the luminance channel intact results in less visible distortions (per unit of data embedded) when compared to data embedding in color channels which also contain luminance information. This does not come as surprise; it is commonly known that the contrast sensitivity of the HVS is higher for the luminance channel than for the red-green or blue-yellow opponent-color channels.[28–31]

In this work, we use the $\mathrm{YC_bC_r}$ opponent-color space as error minimization space for the CDBS algorithm. Using this color space resulted in finest halftone textures when compared to the CIELAB and $\mathrm{Y_yC_xC_z}$ opponent-color spaces. For the luminance channel Y we use the same point-spread function which was proposed in the original DBS/CDBS; it is derived from the luminance contrast sensitivity function proposed by Nasanen.[32] For the point-spread function used for the two chrominance channels C_b and C_r, we start with a Gaussian function[23,33] and add two parameters to control its orientation and eccentricity:

$$\begin{gathered} p_{c,\theta}(x,y) = e^{-(ax_r^2+by_r^2)}, \\ x_r = x\cos(\theta) - y\sin(\theta),\; y_r = y\cos(\theta) + x\sin(\theta), \\ a = \frac{\eta}{2\sigma^2}, \qquad b = \frac{1}{\eta 2\sigma^2}. \end{gathered} \tag{10}$$

In the last equation, θ is angle which controls the orientation $(\theta + 90^\circ)$ of the point-spread function, σ controls its effective spatial support, while x_r and y_r are rotated coordinates. The parameter η $(\eta > 1)$ is a quality factor which controls the eccentricity of the elliptic point-spread function $p_{c,\theta}(x,y)$. The number of possible orientations is 2^B, where B is the amount of information bits which can be embedded in a block of one chrominance channel. The angle difference between neighboring orientations is $180/2^B$ deg. During the CDBS initialization, a separate $\boldsymbol{c}_{pp,i,\theta}[\mathbf{m}]$ matrix is calculated using Eq. (6) for each of the differently oriented point-spread functions. The actual choice of $\boldsymbol{c}_{pp,i,\theta}[\mathbf{m}]$ during minimization is made on a $M \times N$ block basis,

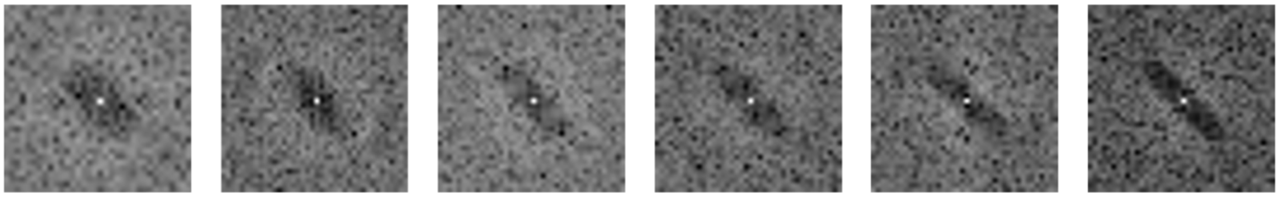

Figure 1. Fourier transform of a Cb halftone block, $\theta = 135°$. The quality factor from left to right: $\eta = 1.5$; 2; 2.5; 3; 3.5; 4.

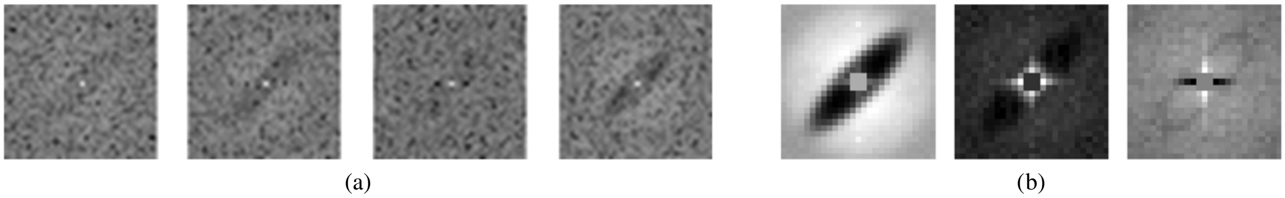

Figure 2. (a) Different variations of embedded orientation feature for $\theta = 45°$, $\eta = 3$. (b) The first three principal components for $\theta = 45°$, $\eta = 3$.

depending on the chrominance channel and the actual data to be embedded.

In contrast to the originally proposed approach,[23] the error minimization is not performed in parallel for all the blocks, but in the usual raster-scan order. The update of an accepted change using Eq. (9) causes changes within the support of $\boldsymbol{c}_{pp,i,\theta}[\mathbf{m}]$ which may influence the neighboring $M \times N$ blocks. However, this influence is such that it ensures smooth transitions between different $M \times N$ blocks. The reason we are using single-pass raster-scan minimization is to prevent generating border artefacts because of using different $\boldsymbol{c}_{pp,i,\theta}[\mathbf{m}]$ through the halftone image, which results in a visually higher quality image.

When increasing the parameter η, the eccentricity of the point-spread function also increases, which forces the error smoothing in the particular orientation to be more localized and emphasized. This practically regulates both the shape of the embedded oriented feature and the distortion in the halftone texture. Figure 1 shows the Fourier transform of different C_b halftone blocks (derived from the same continuous block) for different values of the quality factor η. Please note that while the orientation of the embedded feature is θ, the orientation of the corresponding point-spread function is $\theta + 90°$.

DATA EXTRACTION FROM SCANNED CDBS PRINTS

This section presents the proposed detection scheme for data extraction from scanned CDBS prints. There are few different approaches for detection of oriented features in the state-of-the-art halftoning and data hiding techniques. Guo et al. in the grayscale version of orientation modulation[23] use a trained spatial filter to obtain a single correlation measure for each orientation, which can then be used in a probabilistic channel model for naive Bayesian classification. Bulan et al. use second-order image moments as features for Bayesian classification of orientations in clustered-dot halftones.[15] Son et al. proposed another detector for oriented clustered dots which is employing separate learned dictionary of templates for each orientation.[34] The dictionary which yields to minimal error when representing (with limited number of templates/words) the oriented clustered-dot defines the extracted bit. This approach achieved better results when compared to second-order image moments-based detection, which is expected as a large learned dictionary is able to capture the different variations among clustered dots with same orientation. However, this approach has also proven to be very slow (compared to the moments-based detection it is around 10^3 times slower)[34] because of the computationally expensive search for the set of optimal templates in the dictionary.

In this work we propose a detection method which can be seen as a combination between using correlation-based output of trained filters, and employing different filters which can better capture the variation among oriented CDBS halftone blocks. The data embedding described in Data Embedding using Orientation Modulation is not explicit and direct, but is rather a product of "enforcing certain error distribution" in the converging CDBS algorithm. The speed of convergence is dependent on the image content (and the initial halftone), so a certain energy level of embedded data cannot be strictly guaranteed. This results in variation among the embedded oriented features. It needs to be noted that while the quality factor η regulates, or more accurate, influences the shape of the embedded orientation feature, there is still significant variation for fixed η among the oriented features, which is affected by the different visual content. This feature variation is depicted in Figure 2(a).

In order to account for the oriented feature variation among each orientation, we use principal component analysis (PCA) to learn the main modes of variation. The principal components (eigen-templates) $e_{\theta,k}[\mathbf{m}]$ are calculated for each orientation θ from a digital watermarked halftone, so they are not implicitly printer/scanner dependent. Fig. 2(b) shows the first three principal components obtained using 8192 visually diverse training halftone $M \times N$ templates.

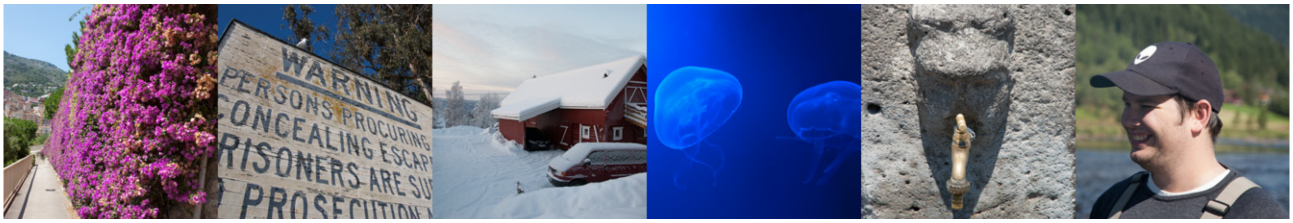

Figure 3. Images used for the evaluation, from left to right indexed as 1 to 6.

The $M \times N$ training templates are transformed using fast Fourier transform (FFT). The central 3×3 coefficients (corresponding to the DC coefficient and the lowest spatial frequencies) are removed as they are highly dependent on the image data but are not useful for detection of the orientation. The same case is with the highest spatial frequencies – they just add noise to the training templates and carry no information about the orientation, so all of the FFT coefficients above the empirically chosen $\Omega = 0.7\pi$ normalized frequency are removed from the training templates. Then PCA is performed which results in the eigen-templates $e_{\theta,k}[\mathbf{m}]$.

The similarity measure for a testing FFT template $t[\mathbf{m}]$, for the orientation θ, is defined by:

$$D_\theta = \sum_{k=1}^{K} \sum_{\mathbf{m}} e_{\theta,k}[\mathbf{m}]t[\mathbf{m}] - \sum_{k=1}^{K} \sum_{\mathbf{m}} e_{\theta+90,k}[\mathbf{m}]t[\mathbf{m}], \quad (11)$$

where D_θ is a similarity measure for the testing template $t[\mathbf{m}]$ using the first K eigen-templates for the θ orientation. The first term on the right side in Equation (11) is clearly a correlation measure between the template $t[\mathbf{m}]$ and the eigen-templates for the θ orientation. The second term is a correlation measure between the template $t[\mathbf{m}]$ and the eigen-templates which correspond to the perpendicular orientation $\theta + 90°$. The fact that two features with perpendicular orientations are the most dissimilar can be observed by the strong negative value of their cross-correlation. Subtracting the two terms in Equation (11) means that the D_θ measure consists of similarity with the inspected orientation θ but also dissimilarity with the perpendicular orientation $\theta + 90°$. The detected orientation θ_d is:

$$\theta_d = \arg\max_{\theta} D_\theta. \quad (12)$$

The detected orientation θ_d is further decoded to the binary bit (or bit-string) it represents. Prior to detection, the scanned image is manually aligned and resized to match the size of the printed halftone. Thus, the geometric distortions introduced during the scanning process (translation, rotation, and scaling) are manually compensated, as an automatic geometric synchronization is not in the focus of this work.

EVALUATION

In this section, we present experimental results for the proposed data hiding technique. The evaluation image set consisted of six images from the publicly available CID:IQ image database.[35] The six images were selected to include wide visual diversity in terms of luminance level, saturation, and texture. The size of all images is 800 × 800. The selected image set is shown in Figure 3. The CDBS halftoning and data hiding algorithm was implemented in C/C++. A channel-independent DBS is calculated first using random initial color halftone; then it is followed by the proposed joint CDBS and data embedding. The total computational time for embedding around 6 kilobits of data in a 1-megapixel halftone (starting from a random halftone) is around 4.3 seconds on the i7-5600U CPU. We printed our CMY halftones at 300 dpi and 600 dpi using the HP Design Jet Z3200 printer and the Caldera raster image processing software. HP Premium Instant-Dry Satin Photo Paper 260 g/m^2 was used, the amount of ink for each of the CMY colorants was set to the experimentally obtained 33% which, for the used photo paper, ensures 100% ink coverage per colorant. The dot-gain compensation curve was approximated using a gamma function ($\gamma = 0.6$ for all colorants). The prints were scanned at twice the printing resolution, using the Epson 10000XL scanner. All color management options were switched off during printing and scanning. Only the first two principal components ($K = 2$) were used for detection, the size of the blocks in the CDBS search strategy is $D = 4$, and the standard deviation in the chrominance point-spread function was set to $\sigma = 2$. An MS Windows demo application for watermark embedding, along with Matlab scripts for watermark detection, can be downloaded from here: http://www.colourlab.no/cid.

Objective Quality Evaluation

An example of both CDBS and watermarked CDBS digital RGB halftone is shown in Figure 4; the quality factor is $\eta = 2.5$, and the difference between the two halftones is difficult to notice when displayed (printed) at 300 dpi or higher resolution. We use two different image quality metrics to provide objective evaluation of the distortion introduced from data embedding. The first metric, HPSNR, is also used in the grayscale version of orientation modulation,[23] so it is using only the luminance information:

$$\text{HPSNR}{=}10\log_{10}\left(\frac{R \times C \times 255^2}{\sum_{\mathbf{m}}(f_Y[\mathbf{m}] * h[\mathbf{m}] - g_Y[\mathbf{m}] * h[\mathbf{m}])^2}\right). \quad (13)$$

In the last equation, R and C denote the number of rows and columns of the image, $f_Y[\mathbf{m}]$ and $g_Y[\mathbf{m}]$ are the luminance

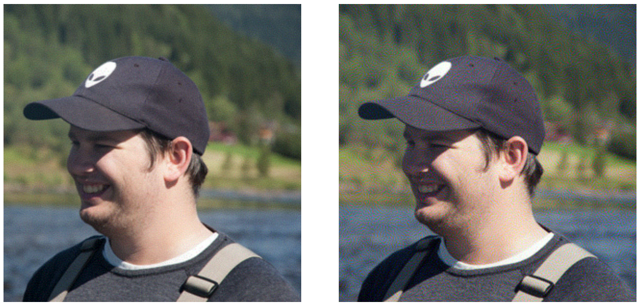

Figure 4. (left) Non-watermarked CDBS RGB digital halftone. (right) Watermarked CDBS RGB digital halftone using blocks of size 32×32, $B = 3$ bits per block, and quality factor $\eta = 2.5$.

Table I. HPSNR for different η.

	$\eta = 1.5$	$\eta = 2.5$	$\eta = 3.5$
HPSNR [dB]	39.8	39.67	39.5

components of the original digital RGB continuous-tone $f[\mathbf{m}]$ and the watermarked RGB halftoned image $g[\mathbf{m}]$, respectively. The symbol '$*$' denotes convolution with a 7×7 normalized Gaussian filter $h[\mathbf{m}]$ with standard deviations $\sigma_x = \sigma_y = 1.3$ (as used in the reference grayscale method).[23] The purpose of this Gaussian filtering is to simulate the low-pass nature of the HVS when calculating the image difference. The results shown in Table I are averages for all of the six test images. It can be seen that the HPSNR values do not vary significantly when increasing the watermark quality factor η (increasing the introduced distortion). Even though different images were used for evaluation in the reference grayscale method, it is worth noting that the obtained average HPSNR values with our method are at least 3.5 dB higher. This shows that, as intended, most of the data embedding distortion is distributed in the chrominance channels.

We use the S-CIELAB metric[36] to measure the perceived color difference ΔE_{ab} between the continuous-tone and the watermarked halftone in digital domain. S-CIELAB is employing spatial filtering to model the HVS behavior, so when calculating the metric, the viewing distance was set to 50 cm, and the halftone pixel resolution was set to both 300 dpi and 600 dpi to match the actual prints. The averaged results for the six test images are shown in Table II. It can be seen that the impact of increasing η is bigger on the 300 dpi halftone than on the 600 dpi halftone. The obtained low ΔE_{ab} values using digital images are consistent with the actual physical prints—it is very difficult to notice the introduced image distortion in the printed images, especially for the 600 dpi prints, regardless of the different values for the quality factor η.

Correct Detection Rate

The CDR is defined as percentage of correctly extracted bits from the scanned prints. When quaternary ($B = 2$) or 8-ary ($B = 3$) symbols are embedded, Gray coding[37] is used to map the n-ary symbols (orientations) to binary strings.

Table II. S-CIELAB metric: ΔE_{ab} for different η.

	$\eta = 1.5$	$\eta = 2.5$	$\eta = 3.5$
300 dpi	0.85	1.09	1.38
600 dpi	0.48	0.53	0.56

Table III. CDR [%] for different block sizes $M \times N$. ($B = 2$, $\eta = 2$).

	24 x 24	32 x 32	40 x 40
300 dpi	87.7	93.2	93.7
600 dpi	86.4	92.7	93.0

We provide results about the dependence of CDR on the block size $M \times N$ used for data embedding, as well as on the different quality factors η, and different number of bits B embedded per chrominance channel block. The scanning resolution is twice the printing resolution, and the scans are manually aligned and resized prior to detection. Table III shows the impact of different block sizes on the CDR. Two bits were embedded per block ($B = 2$), and the quality factor was set to $\eta = 2$. The CDR is averaged for all six images and two chrominance channels. While the CDR for 24×24 blocks is considerably lower, it is almost identical for 32×32 and 40×40 blocks. In all of the remaining results we used 32×32 blocks. It needs to be noted that the 4th image (with significant amount of saturated blue and no red channel at all) contributed most to dropping the total CDR, as the CDR for data embedded in the red-difference channel of this image is around 40–50%. When the result for this image is excluded from the averaging, the overall CDR in Table III is higher for approximately 4 percentage points. Table IV shows the impact of different η and B on the CDR. The shown CDR is averaged for all six test images and chrominance channels. When increasing the quality factor η, the embedded features are more distinctive, so the CDR also increases. When increasing the embedded bits per chrominance channel block B, the CDR decreases as the angle between the embedded orientations becomes smaller. It can be seen that the embedded information is robust for both $B = 1$ and $B = 2$. The same notice about the 4th test image applies to these results as well—if the 4th test image is excluded from the results then the CDR in Table IV will be higher for about 3–4% points. The most common source of detection error is high frequency noise, as well as extreme levels in the color channels (especially the red and the blue channel). When compared to the grayscale method, it can be seen that a significant improvement of CDR is achieved with our proposed method; e.g. the CDR of the method[23] for $B = 2$, and for 300 dpi and 600 dpi prints is around 85% and 55%, respectively. We are unable to provide exact comparison as different test images are used (grayscale versus color).

We tested the CDR dependence on luminance level, by printing a ramp for all 256 gray levels where each gray level

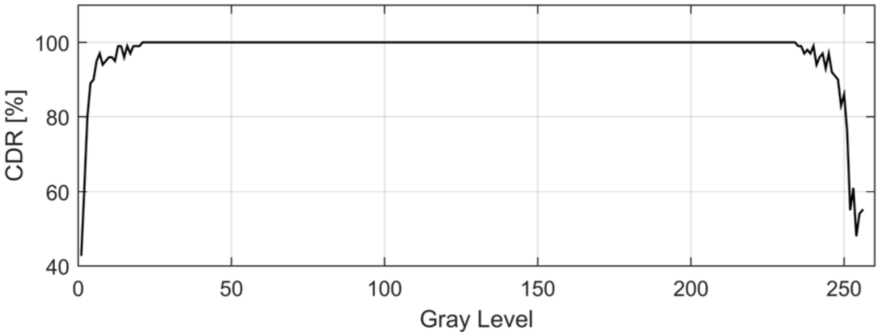

Figure 5. Average CDR (%) for flat gray ($C = M = Y$) patches.

Table IV. CDR [%] for different quality factors and different number of embedded bits per 32 × 32 block, for both 300 dpi and 600 dpi prints of all six test images.

	300 dpi			600 dpi		
	$B = 1$	$B = 2$	$B = 3$	$B = 1$	$B = 2$	$B = 3$
$\eta = 1.5$	94.8	90.3	61.6	93.9	87.8	56.9
$\eta = 2.5$	95.7	94.5	79.2	95.5	93.5	71.9
$\eta = 3.5$	95.9	94.5	83.6	95.7	94.2	79.1

Table V. Maximum number of embeddable bits for minimum CDR = 95%, without using ECC, for different printing resolutions and printed area.

	Square inch	Passport photo (4.5 cm × 3.5 cm)	Business card (9 cm × 5 cm)
300 dpi	350	850	2450
600 dpi	1400	3400	9800

patch contains 100 embeddable blocks. The CDR is shown in Figure 5, and is 100% in the interval [23, 234]. It needs to be noted that this excellent robustness to luminance levels is partly because of the constant (flat) patches used, which ensures stable presence of all three colorants and leads to nearly optimal features embedding and detection.

Data capacity

The total number of bits which can be embedded in a color halftone depends on the halftone size, the block size $M \times N$ used for embedding, and the number of bits embedded per block per chrominance channel, B. When compared to the grayscale version, the available data embedding capacity is doubled as two chrominance channels are used. The CDR results showed that the number of correctly recovered bits additionally depend mainly on the image content and the quality factor η. Different data hiding applications may have different requirements on the CDR. For example, if a binary visual pattern is embedded, then it may be visually recognized from as low as 70% of its bits.[8,38] But if sensitive data is embedded, such as names, ID numbers or URLs, then nothing less than CDR = 100% may be acceptable. We also note that error correction coding (ECC) techniques may be used to ensure perfect data recovery from the prints at the expense of reducing the amount of embedded actual useful data. However, in order for ECC techniques to be beneficial, they may still require high CDR of raw data. So, the actual useful data capacity also depends on the minimally required CDR, and on the ECC technique used (if any). Table V shows the maximum number of raw embeddable bits for different printing resolutions and for minimum 95% CDR without using ECC techniques.

CONCLUSION

In this article, we have proposed an extension of the grayscale orientation modulation data hiding technique[23] to the high quality color direct binary search halftones. The data embedding is performed in the chrominance channels of the YC_bC_r color space, by modulating the orientation of the point-spread function during the iterative halftoning error minimization. A robust similarity measure based on eigen-templates is proposed for the computational detection. The proposed method provides high quality of the watermarked image, variable data embedding capacity, and relatively high robustness to the print-and-scan channel. The experimental results show significant improvement over the grayscale method both in terms of available data embedding capacity and CDRs after printing and scanning. The relatively high CDR makes the proposed approach suitable for data hiding applications which require exact data extraction.

The next steps to follow this work will include: searching for other embeddable features suitable for integration in CDBS halftoning while also improving the scalability of the technique in terms of different types of imaging devices which may be used for data extraction, analyzing the visibility of the embedded data and propose data hiding with uniform visibility across different image content, as well as proposing techniques which will enable automatic geometrical synchronization for data extraction from color prints.

REFERENCES

[1] I. Cox, M. Miller, and J. Bloom, *Digital Watermarking* (San Mateo, CA, Morgan Kaufmann, 2002).

[2] A. M. Alattar, "Smart images using Digimarc's watermarking technology," Proc. SPIE **3971**, 264–273 (2000).

[3] G. Sharma, *Digital Color Imaging Handbook* (CRC, Boca Raton, FL, 2003).

[4] K. Solanki, U. Madhow, B. S. Manjunath, S. Chandrasekaran, and I. El-Khalil, "Print and scan resilient data hiding in images," IEEE Trans. Inf. Forensics Secur. **1**, 464–478 (2006).

[5] M. Fu and O. Au, "Data hiding by smart pair toggling for halftone images," *Proc. IEEE Int'l. Conf. on Acoustics, Speech and Signal Processing* (IEEE, Piscataway, NJ, 2000), pp. 2318–2321.

[6] J. Guo, "Improved data hiding in halftone images with cooperating pair toggling human visual system," Int. J. Imaging Syst. Technol. **17**, 328–332 (2008).

[7] M. Fu and O. Au, "Data hiding watermarking for halftone images," IEEE Tran. Image Process. **11**, 477–484 (2002).

[8] S. Pei and J. Guo, "High-capacity data hiding in halftone images using minimal-error bit searching and least-mean square filter," IEEE Trans. Image Process. **15**, 1665–1679 (2006).

[9] S. Pei and J. Guo, "Hybrid pixel-based data hiding and block-based watermarking for error-diffused halftone images," IEEE Trans. Circuits Syst. Video Technol. **13**, 867–884 (2003).

[10] J. F. Jarvis, C. N. Judice, and W. H. Ninke, "A survey of techniques for the display of continuous-tone pictures on bilevel displays," Comput. Graph. Image Process. **5**, 13–40 (1976).

[11] P. Stucki, "MECCA-A multiple-error correcting computation algorithm for bilevel image hardcopy reproduction" IBM Research Report RZ1060 (1981).

[12] B. Oztan and G. Sharma, "Continuous phase-modulated halftones," IEEE Trans. Image Process. **18**, 2718–2734 (2009).

[13] B. Oztan and G. Sharma, "Multiplexed clustered-dot halftone watermarks using bi-directional phase modulation and detection," *Proc. IEEE Int'l. Conf. on Image Processing* (IEEE, Piscataway, NJ, 2010), pp. 981–984.

[14] B. Oztan and G. Sharma, "Per-separation clustered-dot color halftone watermarks: separation estimation based on spatial frequency content," J. Electron. Imaging **19**, 043007 (2010).

[15] O. Bulan, G. Sharma, and V. Monga, "Orientation modulation for data hiding in clustered-dot halftone prints," IEEE Trans. Image Process. **19**, 2070–2084 (2010).

[16] O. Bulan and G. Sharma, "High capacity color barcodes: per channel data encoding via orientation modulation in elliptical dot arrays," IEEE Trans. Image Process. **20**, 1337–1350 (2011).

[17] M. Analoui and J. Allebach, "Model-based halftoning using direct binary search," Proc. SPIE 96–108 (1992).

[18] D. J. Lieberman and J. Allebach, "A dual interpretation for direct binary search and its implications for tone reproduction and texture quality," IEEE Trans. Image Process. **9**, 1950–1963 (2000).

[19] X. Wan, C. Hu, and J. Xu, "A watermarking algorithm for halftone image based on human vision system model," Proc. SPIE **6493**, 64931H (2007).

[20] X. Zhuge and K. Nakano, "Direct binary search based algorithms for image hiding," Int. J. Digit. Content Technol. Appl. **6**, 457–466 (2012).

[21] D. Kacker and J. Allebach, "Joint halftoning and watermarking," IEEE Trans. Signal Process. **51**, 1054–1068 (2003).

[22] J. Guo, C. Su, and Y. Liu, "Texture orientation modulation for halftoning watermarking," *Proc. IEEE Int'l. Conf. on Acoustics, Speech and Signal Processing* (IEEE, Piscataway, NJ, 2012), pp. 1813–1816.

[23] J. Guo, C. Su, H. Lee, and J. Lee, "Oriented modulation for watermarking in direct binary search halftone images," IEEE Trans. Image Process. **21**, 4117–4127 (2012).

[24] J. Guo, G. Lai, K. Wong, and L. Chang, "Progressive halftone watermarking using multilayer table lookup strategy," IEEE Trans. Image Process. **24**, 2009–2024 (2015).

[25] A. U. Agar and J. Allebach, "Model-based color halftoning using direct binary search," IEEE Trans. Image Process. **14**, 1945–1959 (2005).

[26] T. Flohr, B. Kolpatzik, R. Balasubramanian, D. Carrara, C. Bouman, and J. Allebach, "Model-based color image quantization," Proc. SPIE **1913**, 270–281 (1993).

[27] D. Lieberman and J. Allebach, "Efficient model based halftoning using direct binary search," *Proc. IEEE Int'l. Conf. on Image Processing* (IEEE, Piscataway, NJ, 1997), pp. 775–778.

[28] K. T. Mullen, "The contrast sensitivity of human color vision to red-green and blue-yellow chromatic gratings," J. Physiol. **359**, 381–400 (1985).

[29] A. Reed, E. Rogers, and D. James, "Chrominance Watermark for Mobile Applications," Proc. SPIE **7542**, 75420V (2010).

[30] R. Lyons, A. Reed, and J. Stach, "Watermark embedding in optimal color direction," Proc. SPIE **8665**, 866505 (2013).

[31] A. Tremeau and D. Muselet, "Recent trends in color image watermarking," J. Imaging Sci. Technol. **53**, 010201 (2009).

[32] R. Näsänen, "Visibility of halftone dot textures," IEEE Trans. Syst. Man Cybern. **SMC-14**, 920–924 (1984).

[33] G. M. Johnson and M. D. Fairchild, "Darwinism of color image difference models," *Proc. IS&T/SID 9th Color Imaging Conf.* (IS&T, Springfield, VA, 2001), pp. 108–112.

[34] C. Son and H. Choo, "Watermark detection from clustered halftone dots via learned dictionary," Signal Process. **102**, 77–84 (2014).

[35] X. Liu, M. Pedersen, and J. Y. Hardeberg, "CID:IQ—a new image quality database," Image Signal Process. **8509**, 193–202 (2014).

[36] X. Zhang and B. A. Wandell, "A spatial extension of CIELAB for digital color image reproduction," Soc. Inf. Disp. 96 Dig. 731–734 (1996).

[37] M. Gardner, "The binary gray code," *Knotted Doughnuts and Other Mathematical Entertainments* (W. H. Freeman, New York, 1986), chapter 2.

[38] J. Guo, S. Pei, and H. Lee, "Watermarking in halftone images with parity-matched error diffusion," Signal Process. **91**, 126–135 (2011).